REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0205

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPO... TYPE AND DATES COVERED
		15 March 1999 - 14 March 2000

4. TITLE AND SUBTITLE
Platform for Principled Experimentation of Hard Computational Problems

5. FUNDING NUMBERS
F49620-99-1-0195

6. AUTHOR(S)
Carla P. Gomes

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Science Department
Cornell University

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
801 N. Randolph Street, Room 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-99-1-01951

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release.

12b. DISTRIBUTION CODE

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
NOTICE OF TRANSMITTAL DTIC. THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLIC RELEASE LAW AFR 190-12. DISTRIBUTION IS UNLIMITED.

13. ABSTRACT *(Maximum 200 words)*

Our research in the area of computational complexity of combinatorial problems involves intensive use of computational resources. Traditionally, researchers with such compute intensive needs use supercomputers in a time and node sharing fashion. There are some problems with this approach: (a) Supercomputers tend to be very expensive to acquire (millions of dollars) and to maintain. Also the competition used to be quite weak. (b) Because supercomputers are expensive, parts required to repair them are expensive also, thus the maintenance costs are high. (c) A big number of researchers have access to supercomputers competing for CPU time (and occasionally other resources), thus quite often one has to wait weeks until his/her tasks are run. (d) It is expensive to run something on a supercomputer (since supercomputers are themselves expensive), thus quite big grants are required to do anything serious.

14. SUBJECT TERMS

15. NUMBER OF PAGES
4

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Platform for Principled Experimentation of Hard Computational Problems

Final Report
DURIP Grant
F49620-99-1-0195

P.I.
Carla P. Gomes
gomes@cs.cornell.edu
Computer Science Department
Cornell University

1. The problem

Our research in the area of computational complexity of combinatorial problems involves intensive use of computational resources. Traditionally, researchers with such compute intensive needs use supercomputers in a time and node sharing fashion. There are some problems with this approach:

- Supercomputers tend to be very expensive to acquire (millions of dollars) and to maintain. Also the competition used to be quite weak.
- Because supercomputers are expensive, parts required to repair them are expensive also, thus the maintenance costs are high.
- A big number of researchers have access to supercomputers *competing* for CPU time (and occasionally other resources), thus quite often one has to wait weeks until his/her tasks are run.
- It is expensive to run something on a supercomputer (since supercomputers are themselves expensive), thus quite big grants are required to do anything serious.

2. A Possible Approach

Idea: use *off the shelf* hardware and software components to build a *Personal Supercomputer*. There are numerous success stories of such clusters:

- Loky and Avalon at Los Alamos National Laboratory (used for gravitational simulations).

- Hess (oil company) build a cluster with 130.000$ as powerful as the IBM SP2 they were leasing with 2 million $ for 3 years.
- IBM with 17 dual processor Netfinity machines (160.000$ worth of hardware) got the same performance on povray as a 5 million $ Cray machine.
- Hundreds of 5-20 node clusters with prices around tens of thousands of dollars, powerful enough to do interesting things on them.

This solution has a very good price/performance ratio, small maintenance cost (if a node dies, you just replace it), required software for *making* it into a supercomputer is free (Linux, PVM, MPI, etc.) or has low cost (Windows NT, commercial compilers, commercial MPI implementations).

3. The Zippy Cluster

With the funds from AFOSR (Durip Grant) we purchased the ***Zippy cluster***. It consists of:

- Consists of 4 Dell PowerEdge 8530 machines, each with 8 Pentium III Xeon 550 MHz, 1M L2 Cache, 4G RAM, 18G hard drive.
- An 15 inch monitor and a keyboard is linked to all 4 of them through a multiplexer (mostly unused, only for emergencies).
- Connected to each other and to the department network through Fast Ethernet (100 MBS).
- Operating System: Red Hat Linux 6.1 (SMP version), Kernel 2.2.12.
- Current uptime is about 34 days. Rebooted only two times (once due to a miss configuration and the other time due to power shutdown in the whole building).
- Other important software installed: gcc (C, C++, Java, Fortran compiler), pvm (Parallel Virtual Machine), lam (free implementation of MPI). So far is running only free software.
- Fully remote administration

4. Research Performed

The Zipppy cluster has allowed us to perform very interesting and successful experimental research on the computational complexity of combinatorial problems. In particular we have:

- Characterized the backbone of solutions for the quasigroup problem, i.e., the common structure of all the solutions to a problem istance, identifying two types of backbone: forward checking backbone and full backbone;

- Identified a phase transition in backbone – the phase transition coincides with the hard region of the computational cost;

- Compared computational cost considering different encodings, namely CSP encoding, SAT encoding, and LP based encoding;

- Characterized the impact of structure in terms of computational complexity. In particular we identified the role of balancing and filtering.

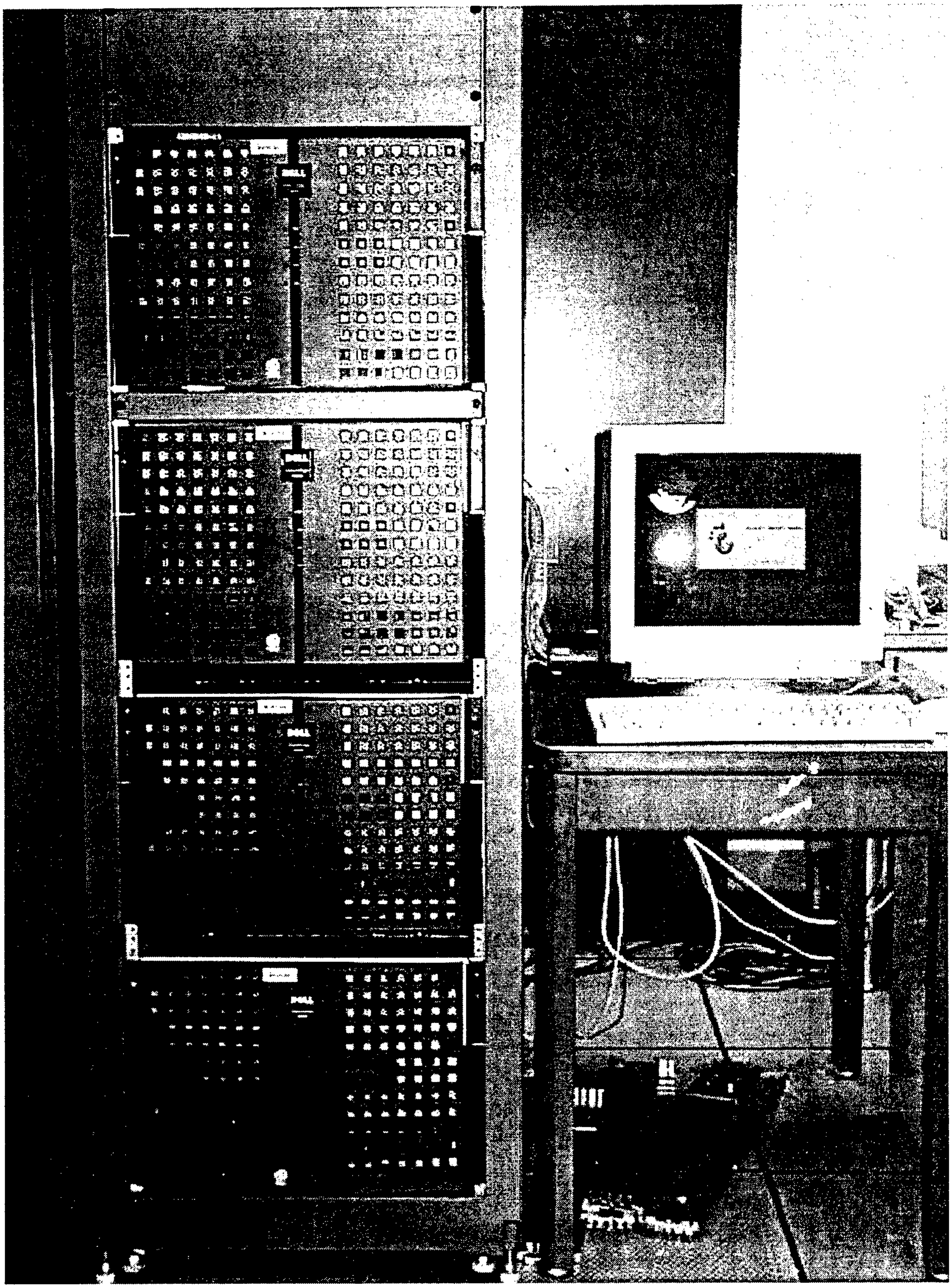
- Developed several benchmark problems for the study of hard combinatorial problems. Of particular interest we have proposed several benchmarks for NP-hard problems that are guaranteed to have at least one solution. We have also identified ways of tuning the hardness of such instances, namely by using balancing and filtering techniques.

5. Future plans

Given the compute intensive nature of our research, our Zippy Cluster is always running at its maximum capacity. We already feel the need to expand it. We would like to:

- Add some more nodes (we would like to acquire another 60 nodes)
- Link the machines between themselves with Gigabit network, maintaining the Fast Ethernet connection with the machines in the department.
- Update to a kernel that uses all the 4G of RAM (currently only 1G is used).
- Install clustering software that will make the 4 machines look like one big virtual machine (support for process migration, task scheduling, load balancing).
- Use some cluster management tools (not needed so far since 4 machines is a manageable number).
- Install some commercial packages (CPLEX, Mathematica, MatLab).

- Parallelize the software we are using (the one we have sources for) using PVM or LAM to take advantage of all the 32 processors in one task if necessary.



•